Compact Labelings For Efficient First-Order Model-Checking

Bruno Courcelle, Cyril Gavoille, Mamadou Moustapha Kanté

Compact Labelings For Efficient First-Order Model-Checking

Bruno Courcelle [1,2] and Cyril Gavoille [1] and Mamadou Moustapha Kanté [1]

Université de Bordeaux, LaBRI, CNRS
351 cours de la Libération
33405 Talence, France

Abstract

We consider graph properties that can be checked from labels, i.e., bit sequences, of logarithmic length attached to vertices. We prove that there exists such a labeling for checking a first-order formula with free set variables in the graphs of every class that is *nicely locally cwd-decomposable*. This notion generalizes that of a *nicely locally tree-decomposable* class. The graphs of such classes can be covered by graphs of bounded *clique-width* with limited overlaps. We also consider such labelings for *bounded* first-order formulas on graph classes of *bounded expansion*. Some of these results are extended to counting queries.

Key words: First-Order Logic; Labeling Scheme; Local Clique-Width; Local Tree-Width; Locally Bounded Clique-Width.

1 Introduction

The model-checking problem for a class of structures $\mathcal{C}$ and a logical language $\mathcal{L}$ consists in deciding for given $S \in \mathcal{C}$, and for some fixed sentence $\varphi \in \mathcal{L}$ if $S \models \varphi$, i.e., if S satisfies the property expressed by φ. More generally, if φ is a formula with free variables $x_1, \ldots, x_m$ one may ask whether S satisfies $\varphi(a_1, \ldots, a_m)$ where $a_1, \ldots, a_m$ are values given to $x_1, \ldots, x_m$. One may also wish to list the set of m-tuples $(a_1, \ldots, a_m)$ that satisfy φ in S, or simply count them.

[1] Supported by the GRAAL project of "Agence Nationale pour la Recherche".
[2] Member of "Institut Universitaire de France".

Polynomial time algorithms for these problems (for fixed φ) exist for certain classes of structures and certain logical languages. In this sense graphs of bounded degree "fit" with first-order (FO for short) logic [28,12] and graphs of bounded tree-width or clique-width "fit" with monadic second-order (MSO for short) logic. Frick and Grohe [14,15,17] have defined *Fixed Parameter Tractable* (FPT for short) algorithms for FO model-checking problems on classes of graphs that may have unbounded degree and tree-width (definitions and examples are given in Section 4) and our results will concern such classes. We will also use graph classes of *bounded expansion*, a notion introduced by Nešetřil and Ossona de Mendez [26].

We will use similar tools for the following labeling problem: let be given a class of graphs $\mathcal{C}$ and a property $P(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ of vertices $x_1, \ldots, x_m$ and of sets of vertices $Y_1, \ldots, Y_q$ of graphs in $\mathcal{C}$. Our aim is to design two algorithms: an algorithm $\mathcal{A}$ that attaches to each vertex x of a given graph of $\mathcal{C}$ a label $L(x)$, defined as a sequence of 0's and 1's, and an algorithm $\mathcal{B}$ that checks the property $P(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ by using the labels and no other information about the considered graph. This latter algorithm must take as input the labels $L(x_1), \ldots, L(x_m)$ and the sets of labels $L(Y_1), \ldots, L(Y_q)$ of the sets $Y_1, \ldots, Y_q$ and tells whether $P(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ is true. Moreover each label $L(x)$ identifies the vertex x in the graph. An *f-labeling scheme* for a class of structures $\mathcal{C}$ is a pair $(\mathcal{A}, \mathcal{B})$ of algorithms solving the labeling problem and using labels of length at most $f(n)$ for n-vertex graphs of $\mathcal{C}$. Results of this type have been established for monadic second-order (MSO for short) logic by Courcelle and Vanicat [10] and, for particular properties (connectivity queries, that are expressible in MSO logic) by Courcelle and Twigg in [9] and by Courcelle et al. in [6].

Let us review the motivations for looking for *compact labelings*. By *compact*, we mean of length of order less than $O(n)$, where n is the number of vertices of the graph, hence in particular of length $\log^{O(1)}(n)$.

In distributed computing over a communication network with underlying graph G, nodes must act according to their local knowledge only. This knowledge can be updated by message passing. Due to space constraints on the local memory of each node, and on the sizes of messages, a distributed task cannot be performing by representing the whole graph G in each node or in each message. It must rather manipulate compact representations of G, distributed in a balanced way over the graph. For an example, the routing task may use routing tables that are sub-linear in the size of G (preferably of poly-logarithmic size), and short addresses transmitted in the headers of messages (of poly-logarithmic size too). As surveyed in [18] many distributed tasks can be optimized by the use of labels attached to vertices. Such labels should be usable even when the network has node or link crashes. They arise from *forbidden-set labeling schemes* in [9]. In this framework, local informations can be updated by transmitting to all surviving nodes the list of (short) labels of all defected nodes and links, so that the surviving nodes can update their local information, e.g., their routing tables.

Let us comment about using set arguments. The forbidden (or defective) parts of a network are handled as sets of vertices passed to a query as an argument. This means that algorithm $\mathcal{A}$ computes the labels once and for all, independently of the possible forbidden parts of the network. In other words the labeling supports node deletions from the given network. (Edge deletions are supported in the labelings of [6] and [9].)

If the network is augmented with new nodes and links, the labels must be recomputed. We leave this incremental extension as a topic for future research.

Set arguments can be used to handle deletions, but also constraints, or queries like "what are the nodes that are at distance at most 3 of X and Y" where X and Y are two specified sets of nodes.

This article is organized as follows. In Section 2 we give some preliminary definitions regarding first-order logic and we define the notions of clique-width and of labeling schemes. Section 3 deals with first-order logic and needed results. In Section 4 we define the notions of *local bounded clique-width* and of *nicely locally cwd-decomposable*. We give some examples and some preliminary results. Section 5 is devoted to the proofs of the main results of this article. In Section 6 we extend some of the main results to counting queries.

2 Definitions

Our results concern graph properties expressed by logical formulas, which assumes that graphs are represented by relational structures. All graphs and relational structures will be finite.

A *relational signature* is a finite set $\mathcal{R} = \{R, S, T, \ldots\}$ of relation symbols, each of which given with an *arity* $ar(R) \geq 1$. A finite relational $\mathcal{R}$-structure S is defined as $\langle D_S, (R_S)_{R\in\mathcal{R}} \rangle$ where $R_S \subseteq D_S^{ar(R)}$. The set D_S is called the *domain* of S. A relational signature $\mathcal{R}$ is binary if $ar(R) \leq 2$ for all $R \in \mathcal{R}$. A relational structure is binary if it is a relational $\mathcal{R}$-structure for some binary relational signature $\mathcal{R}$. We let $\mathcal{R}_i$ be the set of symbols of arity i.

A binary relational $\mathcal{R}$-structure $S = \langle D_S, (R_S)_{R\in\mathcal{R}} \rangle$ will be identified with a colored graph G with vertex set D_S, that has an edge from x to y colored by R in $\mathcal{R}_2$ if and only if $R_S(x, y)$ holds, and such that a vertex x has color P in $\mathcal{R}_1$ if and only if $P(x)$ holds. Hence, G is a directed graph such that each edge has a color and a vertex a possibly empty set of colors [3]. We use standard graph theoretical notations: V_G for vertex set, E_G for edge set and we will write G as the relational structure $\langle\, V_G, (edg_a G)_{a\in C_2}, (p_{aG})_{a\in C_1}\,\rangle$ where $\mathcal{R}_2 = \{edg_a \mid a \in C_2\}$ and $\mathcal{R}_1 = \{p_a \mid a \in C_1\}$. Such a graph is not colored if $\mathcal{R}_2 = \{edg\}$ and $\mathcal{R}_1 = \emptyset$. A graph represented by the relational structure $\langle\, V_G, (edg_a G)_{a\in C_2}, (p_{aG})_{a\in C_1}\,\rangle$ is called a C-graph, $C = C_1 \cup C_2$.

Let G be a graph, colored or not. If X is a subset of the set of vertices of G, we let $G[X]$ be the induced sub-graph of G with vertex set X and induced colors in the obvious way and, we let $G\backslash X$ be the sub-graph $G[V_G - X]$ [4].

If X is a subset of V_G, we let $N_G^t(X)$ be the set $\{y \in V_G \mid d(x,y) \leq t$ for some $x \in X\}$ and $d(x,y)$ is the length of a shortest undirected path between x and y in G.

An undirected graph is a graph G such that $\mathcal{R}_2 = \{edg\}$ and edg is symmetric. If G is any graph and $m \geq 1$, we denote by G^m the simple, loop-free undirected graph such that $V_{G^m} = V_G$ and two distinct vertices x and y are adjacent in G^m if and only if $d(x,y) \leq m$.

A graph has *arboricity* at most k if it is the union of k edge-disjoint forests (independently of the colors of its edges and of its vertices).

We now define *first-order* logic and *monadic second-order* logic on relational structures and thus, on graphs. Let $\mathcal{R}$ be a relational signature. *Atomic* formulas over relational $\mathcal{R}$-structures are $x = y$, $x \in X$ and $R(x_1, \ldots, x_{ar(R)})$ for all relations R in $\mathcal{R}$. A first order formula (FO formula for short) over relational $\mathcal{R}$-structures is a formula formed from atomic formulas over relational $\mathcal{R}$-structures with Boolean connectives $\wedge, \vee, \neg, \Rightarrow$ and first-order quantifications $\exists x$ and $\forall x$. We may have free set variables. A monadic second-order formula (MSO formula for short) over relational $\mathcal{R}$-structures is formed as FO formulas over relational $\mathcal{R}$-structures with set quantifications $\exists X, \forall X$. By formulas (FO or MSO) we mean formulas written with the signature appropriate for the considered relational structures. If the free variables of a formula φ are among $x_1, \ldots, x_m, Y_1, \ldots, Y_q$ we will write $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$. A *sentence* is a formula without free variables. We write $S \models \varphi$ to mean that the sentence φ is satisfied by the relational structure S.

The *tree-width* [4] of a graph is independent of edge directions and of the colors of edges and vertices. It is a well-known graph parameter and yields many algorithmic properties surveyed by Grohe [17] and Kreutzer [23]. The

[3] It is technically useful in many cases to have several colors attached to a vertex. Furthermore, these colored graphs correspond to relational structures with relation symbols of arity 1 (vertex colors) and 2 (edge colors)

[4] If X is the singleton $\{x\}$, we write $G\backslash x$ instead of $G\backslash\{x\}$.

survey [5] by Bodlaender presents tree-width and recent developments about this notion.

Clique-width [8] is another graph parameter that yields interesting algorithmic results. It is sensible to colors and directions of edges. The original definition of clique-width in [8] concerns only uncolored graphs. However, it can be extended to colored graphs [3,13].

Definition 2.1 (Clique-Width of Colored Graphs) *We let C be the finite set ($C = C_1 \cup C_2$) of colors for vertices and edges. In order to construct graphs, we will use the set $[k] := \{1, 2, \ldots, k\}$ for $k \geq 1$ to color also vertices, with one and only one color for each vertex. A k-C-*graph *(or k-*graph *if $\mathcal{R} = \{edg\}$) G is defined as $G = \langle V_G, (edg_{aG})_{a \in C_2}, (p_{aG})_{a \in C_1}, lab_G \rangle$ where $lab_G : V_G \to [k]$ is a total function and the other components are as defined above. We give several operations on k-C-graphs.*

(1) For k-C-graphs G and H such that $V_G \cap V_H = \emptyset$, we let $G \oplus H$ be the k-C-graph K where:

$$V_K = V_G \cup V_H,$$

$$p_{aK}(x) = \begin{cases} p_{aG}(x) & \text{if } x \in V_G \\ p_{aH}(x) & \text{if } x \in V_H \end{cases} \qquad \text{for all } a \in C_1.$$

$$edg_{aK}(x, y) = \begin{cases} edg_{aG}(x, y) & \text{if } x, y \in V_G \\ edg_{aH}(x, y) & \text{if } x, y \in V_H \end{cases} \qquad \text{for all } a \in C_2.$$

$$lab_K(x) = \begin{cases} lab_G(x) & \text{if } x \in V_G \\ lab_H(x) & \text{if } x \in V_H. \end{cases}$$

The graph $G \oplus H$ is well-defined up to isomorphism.

(2) For a k-C-graph G, for a color b in C_2 and for distinct $i, j \in [k]$, we denote by $\eta^b_{i,j}(G)$, the k-C-graph $K = \langle V_G, (edg_{aK})_{a \in C_2}, (p_{aG})_{a \in C_1}, lab_G \rangle$ where:

$$edg_{aK} = \begin{cases} edg_{aG} & \text{if } a \neq b \\ edg_{bG} \cup \{(x, y) \mid x, y \in V_G \ \wedge \ x \neq y \ \wedge \ i = lab_G(x), \ j = lab_G(y)\} & \text{if } a = b. \end{cases}$$

(3) For a k-C-graph G, for distinct $i, j \in [k]$, we denote by $\rho_{i \to j}(G)$, the k-colored graph $K = \langle V_G, (edg_{aG})_{a \in C_2}, (p_{aG})_{a \in C_1}, lab_K \rangle$ where

$$lab_K(x) = \begin{cases} j & \text{if } lab_G(x) = i, \\ lab_G(x) & \text{otherwise.} \end{cases}$$

(4) For each $i \in [k]$ and each $A \subseteq C$, $\mathbf{i_A}$ denotes a k-C-graph with a single vertex x with $lab_{\mathbf{i_A}}(x) = i$ such that $p_{a\mathbf{i_A}}(x)$ holds if and only if $a \in A \cap C_1$

and $edg_{a\mathbf{i_A}}(x,x)$ *holds if and only if* $a \in A \cap C_2$. *We let* $C_{C,k} = \{\mathbf{i_A} \mid i \in [k], A \subseteq C\}$.

We let $F_{C,k} = \{\oplus, \eta^a_{i,j}, \rho_{i\to j} \mid i,j \in [k], a \in C\}$. *Each term* t *in* $T(F_{C,k}, C_{C,k})$ *has a* value $val(t)$*: it is the* k*-*C*-graph obtained by evaluating* t *according to definitions (1)-(4). The clique-width of a (colored) graph* G*, denoted by* $cwd(G)$*, is the minimum* k *such that* G *is isomorphic to* $val(t)$ *for some term* t *in* $T(F_{C,k}, C_{C,k})$.

There is a function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that if a C-graph has tree-width w then, it has clique-width at most $f(w, |C|)$. The proof of [8] that concerns uncolored graphs can be adapted. The converse is false because cliques have clique-width 2 and unbounded tree-width. For fixed k, there exists a cubic-time algorithm that given an undirected C-graph G either outputs that it has clique-width at least $k+1$ or outputs a term t in $T(F^a_{C,k'}, C^a_{C,k'})$ that defines G with $k' = 2^{k+1} - 1$ [27,20]. This algorithm can be adapted to colored graphs with $k' = g(k)$ for some function g [22]. Also, every property expressible in MSO logic can be checked in cubic-time in classes of colored graphs of bounded clique-width by combining the results of [7] and of [22]. The survey by Kamiński et al. [21] presents recent results on clique-width.

We now define the notion of *bounded expansion* [26]. As tree-width, it is independent of colors of vertices and/or edges. Graph classes with *bounded expansion*, defined in [26], have several equivalent characterizations. We will use the following one.

Definition 2.2 (Bounded Expansion) *A class* $\mathcal{C}$ *of colored graphs has* bounded expansion *if for every integer* p*, there exists a constant* $N(\mathcal{C}, p)$ *such that for every* $G \in \mathcal{C}$*, one can partition its vertex set in at most* $N(\mathcal{C}, p)$ *parts such that any* i *parts for* $i \leq p$ *induce a sub-graph of tree-width at most* $i-1$.

The case $i = 1$ of Definition 2.2 implies that each part is a stable set, hence the corresponding partition can be seen as a *proper vertex-coloring*. We finish these preliminary definitions by introducing the notion of *labeling scheme*.

Definition 2.3 (Labeling Scheme) *Let* $\mathcal{R}$ *be a relational signature. Let* $S = \langle D_S, (R_G)_{R\in\mathcal{R}} \rangle$ *be a relational* $\mathcal{R}$*-structure. A* labeling *of* S *is an injective mapping* $J : D_S \to \{0,1\}^*$ *(or into some more convenient set* Σ^* *where* Σ *is a finite alphabet). If* Y *is a subset of* D_S *we let* $J(Y)$ *be the family* $(J(y))_{y\in Y}$. *Clearly each set* Y *is defined from* $J(Y)$.

Let $\varphi(\bar{x}, \overline{Y})$ *be an FO or MSO formula over relational* $\mathcal{R}$*-structures where* $\bar{x}$ *is an* m*-tuple of FO variables and* $\overline{Y}$ *a* q*-tuple of set variables. Let* $\mathcal{C}$ *be a class of relational* $\mathcal{R}$*-structures and let* $f : \mathbb{N} \to \mathbb{N}$ *be an increasing function. An* f-labeling scheme supporting the query *defined by* φ *in the relational* $\mathcal{R}$*-structures of* $\mathcal{C}$ *is a pair* $(\mathcal{A}, \mathcal{B})$ *of algorithms doing the following:*

(1) $\mathcal{A}$ constructs for each S in $\mathcal{C}$ a labeling J of S such that $|J(a)| = O(f(n))$ for every $a \in D_S$, where $n = |D_S|$.

(2) If J is computed from S by $\mathcal{A}$, then $\mathcal{B}$ takes as input an $(m+q)$-tuple $(J(a_1), \ldots, J(a_m), J(W_1), \ldots, J(W_q))$ and says correctly whether:

$$S \models \varphi(\bar{a}, \overline{W}).$$

Labeling schemes based on logical descriptions of queries by MSO formulas have been first defined by Courcelle and Vanicat [10] for graphs of bounded clique-width (whence also of bounded tree-width). We recall this theorem. If $\overline{W}$ is a q-tuple of sets, we let $|\overline{W}| = |W_1| + \cdots + |W_q|$ and if $\bar{a}$ is an m-tuple of vertices, we let $|\bar{a}| = m$.

Theorem 2.4 *Let k be a positive integer and let C be a finite set of colors. Then, for every MSO formula $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ there exists a* log-*labeling scheme $(\mathcal{A}, \mathcal{B})$ for φ on the class of C-graphs of clique-width at most k. Moreover, if the input C-graph has n vertices, algorithm $\mathcal{A}$ computes the labels $J(x)$ of all vertices x in time $O(n^3)$ or in time $O(n \cdot \log(n))$ if the clique-width expression of the graph is given. Given $J(a_1), \ldots, J(a_m)$ and $J(W_1), \ldots, J(W_q)$ algorithm $\mathcal{B}$ checks whether $\varphi(\bar{a}, \overline{W})$ holds in time $O(\log(n) \cdot (|\overline{W}| + |\bar{a}| + 1))$*

For n-vertex C-graphs of tree-width at most k, algorithm $\mathcal{A}$ builds the labelings in time $O(n \cdot \log(n))$.

The proof of Theorem 2.4 combines the construction of [10] that works for graphs given with their decompositions, and "parsing" results by Bodlaender [4] for tree-width and, by Hliněný, Oum and Seymour [20,27] and Kanté [22] for clique-width (discussed above). Labeling schemes for distance and connectivity queries in respectively graphs of bounded clique-width and in planar graphs have been given respectively by Courcelle and Twigg in [9] and by Courcelle, Gavoille, Kanté and Twigg in [6].

In the present article, we consider classes of graphs of unbounded clique-width and, in particular, classes that are *locally decomposable* (Frick and Grohe [14,15]) and classes of bounded expansion. So, MSO logic is out of reach for such classes and we will consider FO logic over C-graphs.

3 Bounded and Local First-Order Formulas

The definitions below concern binary relational structures called graphs since they correspond to colored graphs as explained in Section 2. Formulas are written over binary relational structures for a fixed binary relational signature that we do not specify all the time.

Definition 3.1 (Bounded Formulas) *An FO formula* $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ *is a* basic bounded *formula if for some* $p \in \mathbb{N}$ *we have the following equivalence for all graphs* G*, all* $a_1, \ldots, a_m \in V_G$ *and all* $W_1, \ldots, W_q \subseteq V_G$

$G \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q)$ *iff* $G[X] \models \varphi(a_1, \ldots, a_m, W_1 \cap X, \ldots, W_q \cap X)$

for some $X \subseteq V_G$ *such that* $a_1, \ldots, a_m \in X$ *and* $|X| \leq p$.

If this is true for X*, then* $G[Y] \models \varphi(a_1, \ldots, a_m, W_1 \cap Y, \ldots, W_q \cap Y)$ *for every* $Y \supseteq X$*. We call* p *a* bound on the quantification space*.*

An FO formula is bounded *if it is a Boolean combination of basic bounded formulas.*

The negation of a basic bounded formula is not (in general) basic bounded, but it is bounded. The property that a graph has a sub-graph isomorphic to a fixed graph H is expressible by a bounded formula.

We still call *sentence* an FO formula without free FO variables that has free set variables.

Definition 3.2 (Local Formulas) *An FO formula* $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ *is* t-local around $(x_1, \ldots, x_m)$ *if for every graph* G*, all* $a_1, \ldots, a_m$ *in* V_G *and all subsets* $W_1, \ldots, W_q$ *of* V_G *we have*

$G \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q)$ *iff* $G[N] \models \varphi(a_1, \ldots, a_m, W_1 \cap N, \ldots, W_q \cap N)$

where $N = N^t_G(\{a_1, \ldots, a_m\})$.

An FO sentence $\varphi(Y_1, \ldots, Y_q)$ *is* basic (t, s)-local *if it is equivalent to a sentence of the form*

$$\exists x_1 \cdots \exists x_s \left(\bigwedge_{1 \leq i < j \leq s} d(x_i, x_j) > 2t \ \wedge \bigwedge_{1 \leq i \leq s} \psi(x_i, Y_1, \ldots, Y_q) \right)$$

where $\psi(x, Y_1, \ldots, Y_q)$ *is* t*-local around its unique free variable* x*.*

Remark 3.3 *The property* $d(x, y) \leq r$ *is basic bounded (for* $p = r + 1$*) and* t*-local for* $t = \lfloor r/2 \rfloor$*. Its negation* $d(x, y) > r$ *is* t*-local and bounded (but not basic bounded).*

We now recall a decomposition of *FO* formulas into t-local and basic (t', s)-local formulas due to Gaifman [16].

Theorem 3.4 ([24]) *Every FO formula* $\varphi(\bar{x}, \overline{Y})$ *is logically equivalent to a Boolean combination* $B\left(\varphi_1(\overline{u_1}, \overline{Y}), \ldots, \varphi_p(\overline{u_p}, \overline{Y}),\ \psi_1(\overline{Y}), \ldots, \psi_h(\overline{Y})\right)$ *where:*

- *each φ_i is a t-local formula around some sub-sequence $\overline{u_i}$ of $\bar{x}$,*
- *each ψ_i is a basic (t', s)-local sentence.*

Moreover B can be computed effectively and, the integers t, t' and s can be bounded in terms of m and the quantifier-rank of φ.

This theorem is usually stated and proved for FO formulas without free set variables. However, in an FO formula, a set variable Y_i occurs in atomic formulas of the form "$y \in Y_i$". This is equivalent to "$R_i(y)$" if R_i is a unary relation representing Y_i. We denote by $\varphi'(\bar{x})$ the formula obtained from $\varphi(\bar{x}, Y_1, \ldots, Y_q)$ by replacing every sub-formula "$y \in Y_i$" by "$R_i(y)$". In order to prove that two FO formulas $\varphi(\bar{x}, Y_1, \ldots, Y_q)$ and $\psi(\bar{x}, Y_1, \ldots, Y_q)$ are equivalent in every relational structure of a class $\mathcal{C}$ of relational $\mathcal{R}$-structures, it is enough to prove that the corresponding formulas $\varphi'(\bar{x})$ and $\psi'(\bar{x})$ are equivalent in every relational structure S' that is an expansion of a relational structure S in $\mathcal{C}$ by unary relations $R_1, \ldots, R_q$. Hence, Theorem 3.4 follows from its usual formulation for FO formulas without free set variables. The same holds for Theorem 3.5 below.

We will use a stronger form of Theorem 3.4 from [14], that decomposes t-local formulas. Let $m, t \geq 1$. The *t-distance type* of an m-tuple $\bar{a}$ is the undirected graph $\Delta(\bar{a}) = ([m], edg_{\Delta(\bar{a})})$ where $edg_{\Delta(\bar{a})}(i,j)$ iff $d(a_i, a_j) \leq 2t+1$. For each graph Δ the property that an m-tuple $\bar{a}$ satisfies $\Delta(\bar{a}) = \Delta$ can be expressed by a t-local formula $\rho_{t,\Delta}(x_1, \ldots, x_m)$ equivalent to:

$$\bigwedge_{(i,j)\in edg_\Delta} d(x_i, x_j) \leq 2t+1 \ \wedge \bigwedge_{(i,j)\notin edg_\Delta} d(x_i, x_j) > 2t+1.$$

Theorem 3.5 ([14]) *Let $\varphi(\bar{x}, \overline{Y})$ be a t-local formula around the m-tuple $\bar{x}$, $m \geq 1$ with $\overline{Y} = (Y_1, \ldots, Y_q)$. For each t-distance type Δ with connected components $\Delta_1, \ldots, \Delta_p$ one can compute a Boolean combination $F^{t,\Delta}(\varphi_{1,1}, \ldots, \varphi_{1,j_1}, \ldots, \varphi_{p,1}, \ldots, \varphi_{p,j_p})$ of formulas $\varphi_{i,j}$ with free variables in $\bar{x}$ and in $\overline{Y}$ such that:*

- *The free FO variables of each $\varphi_{i,j}$ belong to $\bar{x} \mid \Delta_i$ (where $\bar{x} \mid \Delta_i$ denotes the restriction of $\bar{x}$ to Δ_i).*
- *$\varphi_{i,j}$ is t-local around $\bar{x} \mid \Delta_i$.*
- *For each m-tuple $\bar{a}$, each q-tuple of sets $\overline{W}$, $G \models \rho_{t,\Delta}(\bar{a}) \wedge \varphi(\bar{a}, \overline{W})$ iff $G \models \rho_{t,\Delta}(\bar{a}) \wedge F^{t,\Delta}(\ldots, \varphi_{i,j}(\bar{a} \mid \Delta_i, \overline{W}), \ldots)$.*

We are interested in on-line checking properties of networks in case of (reported) failures of some nodes (nodes are vertices of the associated graphs). Hence, for each property of interest, defined by a formula $\varphi(x_1, \ldots, x_m)$, we are not only interested in checking if $G \models \varphi(a_1, \ldots, a_m)$ by using $J(a_1), \ldots, J(a_m)$ for $a_1, \ldots, a_m \in V_G$, but also, in checking if $G \backslash W \models \varphi(a_1, \ldots, a_m)$ by using $J(a_1), \ldots, J(a_m)$ and $J(W)$ where W is a subset of $V_G - \{a_1, \ldots, a_m\}$. However, the property $G \backslash W \models \varphi(a_1, \ldots, a_m)$ for an FO formula $\varphi(x_1, \ldots, x_m)$ is equivalent to $G \models \varphi'(a_1, \ldots, a_m, W)$ and to $G_W \models \varphi''(a_1, \ldots, a_m)$ for FO formulas $\varphi'(x_1, \ldots, x_m, Y)$ and $\varphi''(x_1, \ldots, x_m)$ that are easy to write. We denote by G_W the graph G equipped with an additional vertex-color $\perp$, i.e., as the structure G expanded with a unary relation $p_\perp$ such that $p_{\perp G_W}(u)$ holds iff $u \in W$. We will handle "holes" in graphs by means of set variables.

4 Locally Decomposable Classes

We will use the same notations as in [14,15]. Definition 4.1 is analogous to [15, Definition 5.1].

Definition 4.1 (Local Clique-Width)

(1) *The* local clique-width *of a graph G is the function* $lcw^G : \mathbb{N} \to \mathbb{N}$ *defined by* $lcw^G(t) := \max\{cwd(G[N_G^t(a)]) \mid a \in V_G\}$.
(2) *A class* $\mathcal{C}$ *of graphs has* bounded local clique-width *if there is a function* $f : \mathbb{N} \to \mathbb{N}$ *such that* $lcw^G(t) \leq f(t)$ *for every* $G \in \mathcal{C}$ *and* $t \in \mathbb{N}$.

Examples of Graphs of Bounded Local Clique-Width

(1) Every class of graphs of bounded clique-width has also bounded local clique-width since $cwd(G[A]) \leq cwd(G)$ for every $A \subseteq V_G$ (see [8]).
(2) The classes of graphs of bounded local tree-width have bounded local clique-width since every class of graphs of bounded tree-width has bounded clique-width (see [8]). We can cite graphs of bounded degree and minor-closed classes of graphs that exclude some apex-graph as a minor [5] (see [14,15]) as examples of classes of bounded local tree-width.
(3) Let m be a positive integer and let $\mathcal{C}$ be a class of graphs of bounded local clique-width. Then $\mathcal{C}^m = \{G^m \mid G \in \mathcal{C}\}$ has bounded local clique-width. Let sketch the proof. Let G be a graph in $\mathcal{C}$. For every vertex x of G and every positive integer r we have $N^r_{G^m}(x) \subseteq N^{rm}_G(x)$. Hence, for every graph G in $\mathcal{C}$ and for every positive integer r, $lcw^{G^m}(r) \leq f(rm)$ where f is the function that bounds the local clique-width of graphs in $\mathcal{C}$.

    The same holds for $Line(\mathcal{C}) = \{Line(G) \mid G \in \mathcal{C}\}$ [6] if $\mathcal{C}$ has bounded local tree-width. Let G be a graph in $\mathcal{C}$ and let $K = Line(G)$. For every e and e' in $E_G = V_K$ we have $d_G(x, y) \leq d_K(e, e') + 1$ if x is any end vertex of e and y is any end vertex of e'. It follows that $K[N^r_K(e)] = Line(H)$

[5] An *apex-graph* is a graph G such that $G \backslash u$ is planar for some vertex u.
[6] If G is a graph we denote by $K = Line(G)$, called the *line graph* of G, the graph with vertex set the set of edges of G and $edg_K(x, y)$ holds if and only if x and y are incident.

where H is a sub-graph of $G[N_G^{r+1}(x)]$ and x is an end vertex of e. If $\mathcal{C}$ has bounded local tree-width then $twd(H) \leq twd(G[N_G^{r+1}(x)) \leq f(r)$ [7] for some function f, hence $cwd(K[N_K^r(e)]) = cwd(Line(H)) \leq g(f(r))$ for some function g by a result of [19]. Hence, the class $\mathcal{C}$ has bounded local clique-width.

(4) The class of interval graphs has not bounded local clique-width. Otherwise, interval graphs would have bounded clique-width, because if we add to an interval graph a new vertex adjacent to all, we obtain an interval graph of diameter 2.

In order to obtain a log-labeling scheme for certain classes of graphs of bounded local clique-width, we will cover their graphs, as in [14,15], by graphs of bounded clique-width. In [14] a notion of *nicely locally tree-decomposable* class of structures was introduced. We will define a slightly more general notion. But before we define the intersection graph of a *cover* of a graph G, i.e., a family $\mathcal{T}$ of subsets of V_G the union of which is V_G.

Definition 4.2 (Intersection Graph) *Let G be a graph and let $\mathcal{T}$ be a cover of G. The* intersection graph of $\mathcal{T}$ *is the undirected graph $G(\mathcal{T})$ where $V_{G(\mathcal{T})} := \{x_U \mid U \in \mathcal{T}\}$ and $x_U x_V \in E_{G(\mathcal{T})}$ if and only if $U \cap V \neq \emptyset$.*

Definition 4.3 (Graph Covers) *Let $r, \ell \geq 1$ and $g : \mathbb{N} \to \mathbb{N}$. An (r, ℓ, g)-*cwd cover *of a graph G is a family $\mathcal{T}$ of subsets of V_G such that:*

(1) For every $a \in V_G$ there exists a set $U \in \mathcal{T}$ such that $N_G^r(a) \subseteq U$.
(2) The graph $G(\mathcal{T})$ has degree at most ℓ.
(3) For each $U \in \mathcal{T}$ we have $cwd(G[U]) \leq g(1)$.

An (r, ℓ, g)-cwd cover is nice *if condition (3) is replaced by condition (3') below:*

(3') For all $U_1, \ldots, U_q \in \mathcal{T}$ and $q \geq 1$ we have $cwd(G[U_1 \cup \cdots \cup U_q]) \leq g(q)$.

A class $\mathcal{C}$ of graphs is (nicely) locally cwd-decomposable *if every graph G in $\mathcal{C}$ has, for each $r \geq 1$, a (nice) (r, ℓ, g)-cwd cover for some ℓ, g depending on r (but not on G).*

The notions of locally cwd-decomposable and of nicely locally cwd-decomposable are the same as in [15,14] where we substitute clique-width to tree-width except that our definition requires nothing about the time necessary to compute covers.

[7] We denote by $twd(G)$ the tree-width of a graph G.

Examples of (Nicely) Locally Cwd-Decomposable Graph Classes

(1) Every nicely locally cwd-decomposable class is locally cwd-decomposable and the converse does not seem to be true (but we do not have a counterexample).
(2) Each class of nicely locally tree-decomposable graph is nicely locally cwd-decomposable.
(3) We do not know if every graph class of bounded local clique-width is locally cwd-decomposable. We conjecture that there exists a graph class of bounded local clique-width which is not locally cwd-decomposable.
(4) Figure 1 shows inclusion relations between the many classes defined in Sections 3 and 4.

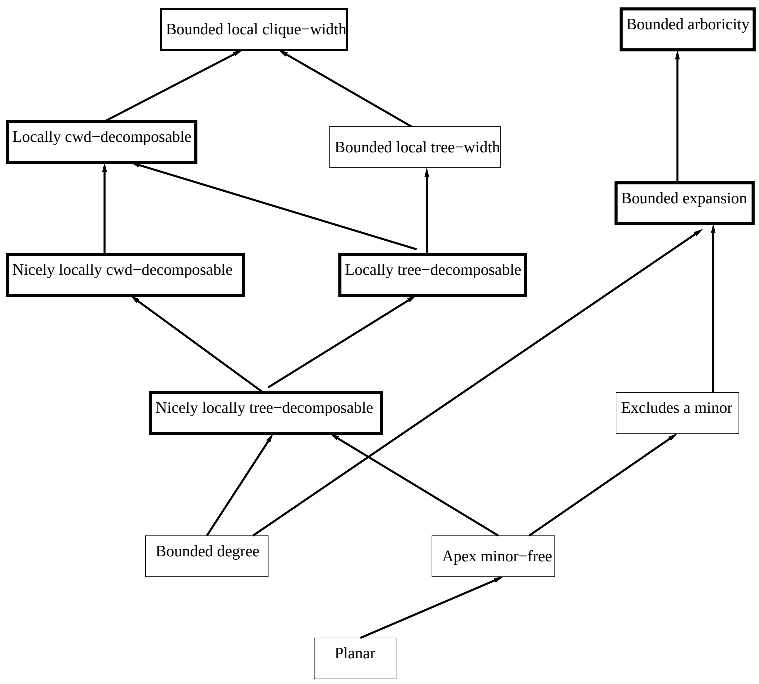

Fig. 1. Inclusion diagram indicating which results apply to which classes. An arrow means an inclusion of classes. Bold boxes are used in this paper.

Fact 4.4 *The class of unit-interval graphs is nicely locally cwd-decomposable.*

Proof. We first prove that unit-interval graphs have bounded local clique-width. We let $H_{n,m}$ be the graph $\langle V_1 \cup \cdots \cup V_n, E_1 \cup E_2 \rangle$ with nm vertices such that:

$$\begin{aligned} V_i &= \{v_{i,1}, \ldots, v_{i,m}\}, \\ E_1 &= \bigcup_{1 \leq i \leq n} \{v_{i,j}v_{i,\ell} \mid j, \ell \leq m\}, \\ E_2 &= \bigcup_{\substack{1 \leq i \leq n-1 \\ 1 \leq j \leq m}} \{v_{i,j}v_{\ell,j} \mid \ell = i+1, \ldots, m\} \end{aligned}$$

Figure 2 shows the graph $H_{4,4}$. Lozin [25] showed that every unit-interval graph with n vertices is an induced sub-graph of $H_{n,n}$.

Let G be a unit-interval graph with n vertices. Then for every positive integer r and every vertex x of G the sub-graph $G[N_G^r(x)]$ is an induced sub-graph of $H_{r,n}$, i.e., has clique-width at most $3r$ since for every positive integers s and t the clique-width of $H_{s,t}$ is at most $3s$ [25]. (Bagan gives in [2] another proof stating that unit-interval graphs have bounded local clique-width.)

We now prove that the class of unit-interval graphs is nicely locally cwd-decomposable. Let G be a unit-interval graph. For $1 \leq i \leq n-1$ we let $G_i = N_G^{r+1}(v_{i,1})$. It is clear that the family $\{G_i \mid 1 \leq i \leq n-1\}$ is a nice $(r, 2r+2, 3 \cdot (r+1))$-cwd cover of G. $\square$

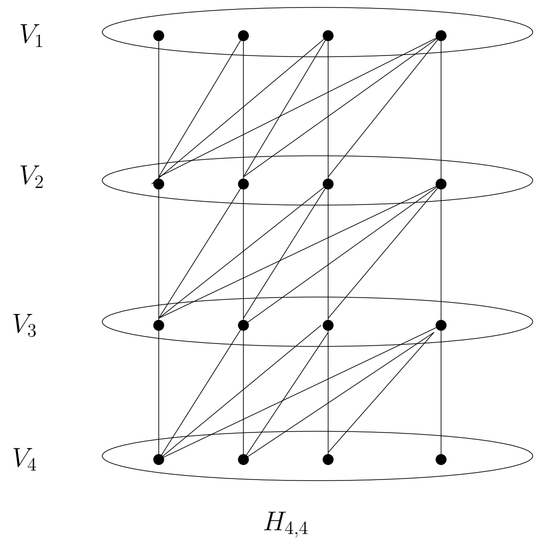

Fig. 2. The graph $H_{4,4}$. Each V_i, for $1 \leq i \leq 4$, induces a clique.

The lemma below is an easy adaptation of the results in [15].

Lemma 4.5 *Let G be in a class of graphs of bounded local clique-width and let φ be a basic (t, s)-local sentence without set variables. We can check in time $O(n^4)$ whether G satisfies φ, $n = |V_G|$.*

Proof Sketch. Let G be in a class $\mathcal{C}$ of graphs of bounded local clique-width and let f be the function that bounds the local clique-width of graphs in $\mathcal{C}$. Let φ be a basic (t,s)-local sentence, equivalent to

$$\exists x_1 \cdots \exists x_s \left(\bigwedge_{1 \leq i < j \leq s} d\left(x_i, x_j\right) > 2t \ \wedge \bigwedge_{1 \leq i \leq s} \psi(x_i) \right)$$

where $\psi(x)$ is t-local around its unique free variable x.

For each vertex a in G we can compute the set $N_G^t(a)$, of size at most n, in time $O(n^2)$. Since $cwd(G[N_G^t(a)]) \leq f(t)$, we can verify in time $O(n^3)$ if G satisfies $\varphi(a)$ by combining the results in [20] and in [7]. We can then compute in time $O(n^4)$ the set $\{a \in V_G \mid G \models \varphi(a)\}$. The formula φ is valid in G if and only if there exist $a_1, \ldots, a_s$ in P such that $d(a_i, a_j) > 2t$. It is proved in [17] that we can verify their existence in time $O(n^3)$. $\square$

5 Labeling Schemes for First-Order Queries

Our results concern 4 types of graph classes (see Figure 1) and 5 types of FO queries. We now state the main theorem of this section.

Theorem 5.1 (First Main Theorem) *There exist* $\log$-*labeling schemes* $(\mathcal{A}, \mathcal{B})$ *for the following queries and graph classes. In each case the input graph has* n *vertices and each query is denoted by* $\varphi(\bar{x}, \overline{Y})$.

(1) Quantifier-free queries in graphs of bounded arboricity. Algorithm $\mathcal{A}$ *constructs a labeling in time* $O(n)$. *Algorithm* $\mathcal{B}$ *gives the answer in time* $O(\log(n) \cdot (|\bar{a}| + |\overline{W}| + 1))$ *for every tuples* $\bar{a}$ *and* $\overline{W}$. *The same labeling can be used to check all quantifier-free queries.*

(2) Bounded FO queries for each class of graphs of bounded expansion. Algorithm $\mathcal{A}$ *constructs a labeling in time* $O(n)$. *Algorithm* $\mathcal{B}$ *gives the answer in time* $O(\log(n) \cdot (|\bar{a}| + |\overline{W}| + 1))$ *for every tuples* $\bar{a}$ *and* $\overline{W}$.

(3) Local queries with set arguments on locally cwd-decomposable classes. Algorithm $\mathcal{A}$ *constructs a labeling in time* $O(f(n) + n^4)$ *where* f *is the time taken to construct a cwd-cover. Algorithm* $\mathcal{B}$ *gives the answer in time* $O(\log(n) \cdot (|\bar{a}|^2 + |\overline{W}| + 1))$ *for every tuples* $\bar{a}$ *and* $\overline{W}$.

(4) FO queries without set arguments on locally cwd-decomposable classes. Algorithm $\mathcal{A}$ *constructs a labeling in time* $O(f(n) + n^4)$ *where* f *is the time taken to construct a cwd-cover. Algorithm* $\mathcal{B}$ *gives the answer in time* $O(\log(n) \cdot (|\bar{a}|^2))$ *for every tuple* $\bar{a}$.

(5) FO queries with set arguments on nicely locally cwd-decomposable classes. Algorithm $\mathcal{A}$ *constructs a labeling in time* $O(f(n) + n^4)$ *where* f *is the time taken to construct a nice cwd-cover. Algorithm* $\mathcal{B}$ *gives the answer*

in time $O(\log(n) \cdot (|\bar{a}|^2 + |\overline{W}| + 1))$ *for every tuples* $\bar{a}$ *and* $\overline{W}$.

Proof of Theorem 5.1 (1). Let G be a colored graph with n vertices, represented by the relational structure $\langle V_G, (edg_{aG})_{a \in C_2}, (p_{aG})_{a \in C_1} \rangle$. We recall that edg_a is binary and p_a is unary.

Assume that $und(G)$, the graph obtained from G by forgetting edge directions and colors of vertices and of edges, is a forest. Let R be a subset of V_G that contains one and only one vertex of each connected component, which is a tree, of G. For each color a in C_2 we let $f_a^+, f_a^- : V_G \to V_G$ be mappings such that:

- $f_a^+(u) = v$ iff $edg_a(u, v)$ in G and v is on the unique undirected path between u and some vertex of R
- $f_a^-(u) = v$ iff $edg_a(v, u)$ in G and v is on the unique undirected path between u and some vertex of R.

The edge relation in G is defined by:

$$edg_{aG}(u, v) \Longleftrightarrow \; v = f_a^+(u) \vee \; u = f_a^-(v) \tag{1}$$

If G is the union of k edge-disjoint forests $F_1, \ldots, F_k$ we take the pairs $(f_{i,a}^+, f_{i,a}^-)$ for each forest $und(F_i)$. The edge relation of G is defined in a similar way as in (1) with $2k$ unary functions by letting

$$edg_{aG}(u, v) \Longleftrightarrow \bigvee_{i \in [k]} v = f_{i,a}^+(u) \vee \; u = f_{i,a}^-(v) \tag{2}$$

We let $C_1 = \{c_1, \ldots, c_\ell\}$. For each vertex x of G we let b_x be the Boolean vector $(b_{a_1}, \ldots, b_{a_\ell})$ where $b_{a_i} = 1$ if and only if $p_{c_iG}(x)$ holds. If vertices are numbered from 1 to n and $\ulcorner x \urcorner$ is the bit representation of the index of x, then we let

$$J(x) = \Big(\ulcorner x \urcorner, \ulcorner f_{1,a_1}^+(x) \urcorner, \ulcorner f_{1,a_1}^-(x) \urcorner, \ldots, \ulcorner f_{k,a_\ell}^+(x) \urcorner, \ulcorner f_{k,a_\ell}^-(x) \urcorner, b_x\Big).$$

It is clear that $|J(x)| = O(\log(n))$. We now explain how to check any quantifier-free formula.

Let $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be a quantifier-free formula. For all m-tuples $(a_1, \ldots, a_m)$ of V_G and all q-tuples $(W_1, \ldots, W_q)$ of subsets of V_G we can determine $G[\{x_1, \ldots, x_m\} \cup W_1 \cup \cdots \cup W_q]$ from $J(a_1), \ldots, J(a_m)$ and $J(W_1), \ldots, J(W_q)$, and check if $\varphi(\bar{a}, \overline{W})$ holds.

It is clear that if the input graph has n vertices and m edges then our algorithm constructs the labels in time $O(n+m)$. But, if a graph G has arboricity at most

k, then the number of edges is linear in the number of vertices of G. Therefore, the labels are constructed in linear-time. We now examine the time taken to check whether G satisfies $\varphi(a_1, \dots, a_m, W_1, \dots, W_q)$. For each $x \in \{a_1, \dots, a_m\}$ it takes constant time to check whether $p_{c_iG}(x)$ holds by using the b_x part of $J(x)$. For every x and y in $W_1 \cup \dots \cup W_q \cup \{a_1, \dots, a_m\}$ and every c in C_2 it takes time $O(\log(n))$ to check whether $edg_{cG}(x, y)$ holds and it takes time $O(|W_i| \cdot \log(n))$ to check if x is in W_i. Therefore we can check the validity of $\varphi(a_1, \dots, a_m, W_1, \dots, W_q)$ in time $O(\log(n) \cdot (|\overline{W}| + |\bar{a}| + 1))$ since a quantifier-free formula is a Boolean combination of atomic formulas. $\square$

Proof of Theorem 5.1 (2). Let $\mathcal{C}$ be a class of graphs of bounded expansion and let G in $\mathcal{C}$ be a graph with n vertices, represented by the relational structure $\langle V_G, (edg_{aG})_{a \in C_2}, (p_{aG})_{a \in C_1} \rangle$. Let $\varphi(x_1, \dots, x_m, Y_1, \dots, Y_q)$ with $m \geq 1$ be a be a basic bounded formula with bound p on the quantification space (see Definition 3.1). We let $N = N(\mathcal{C}, p)$ and we partition V_G into $V_1 \uplus V_2 \uplus \dots \uplus V_N$ as in the definition (Definition 2.2) with each V_i nonempty. (We denote by $\uplus$ the disjoint union of sets.)

For every $\alpha \subseteq [N]$ of size p we let $V_\alpha = \bigcup_{i \in \alpha} V_i$ so that the tree-width of $G[V_\alpha]$ is at most $p - 1$. Each vertex x belongs to less than $(N-1)^{p-1}$ sets V_α.

Hence the basic bounded formula $\varphi(\bar{x}, \overline{Y})$ is true in G iff it is true in some $G[X]$ with $|X| \leq p$, hence in some $G[V_\alpha]$ such that $x_1, \dots, x_m \in V_\alpha$. For each α we construct a labeling J_α of $G[V_\alpha]$ (of tree-width at most $p-1$) supporting query φ by using Theorem 2.4. We let $J(x) = \big(\ulcorner x \urcorner, \{(\ulcorner \alpha \urcorner, J_\alpha(x)) \mid x \in V_\alpha\}\big)$. We have $|J(x)| = O(\log(n))$.

Given vertices $a_1, \dots, a_m$ and sets of vertices $W_1, \dots, W_q$ we now explain how to decide the validity of $\varphi(\bar{a}, \overline{Y})$ by using $J(a_1), \dots, J(a_m)$ and $J(W_1), \dots, J(W_q)$. From $J(a_1), \dots, J(a_m)$ we can determine all those sets α such that V_α contains $a_1, \dots, a_m$. Using the components $J_\alpha(\cdot)$ of $J(a_1), \dots, J(a_m)$ and the labels in $J(W_1), \dots, J(W_q)$ we can determine if for some α, $G[V_\alpha] \models \varphi(a_1, \dots, a_m, W_1 \cap V_\alpha, \dots, W_q \cap V_\alpha)$ hence whether $G \models \varphi(a_1, \dots, a_m, W_1, \dots, W_q)$.

It remains to consider the case of a basic bounded formula of the form $\varphi(Y_1, \dots, Y_q)$, i.e., where $m = 0$. For each α we determine the truth value b_α of $\varphi(\emptyset, \dots, \emptyset)$ in $G[V_\alpha]$. The family of pairs (α, b_α) is of fixed size (depending on p) and is appended to $J(x)$ defined as above (suitably appended as a sequence of bits). From $J(W_1), \dots, J(W_q)$ we get $D = \{\alpha \mid V_\alpha \cap (W_1 \cup \dots \cup W_q) \neq \emptyset\}$.

By using the $J_\alpha(\cdot)$ components of the labels in $J(W_1) \cup \dots \cup J(W_q)$ we can determine if for some $\alpha \in D$ we have $G[V_\alpha] \models \varphi(W_1 \cap V_\alpha, \dots, W_q \cap V_\alpha)$. If one is found we can conclude positively. Otherwise, we look for some $b_\beta =$ TRUE such that $\beta \notin D$. The final answer is positive if such β is found.

For a Boolean combination of basic bounded formulas $\varphi_1, \dots, \varphi_t$ with associated labelings $J_1, \dots, J_t$ we take the concatenation $J_1(x), J_2(x), \cdots, J_t(x)$ of the corresponding labels. It is of size $O(\log(n))$ and gives the desired result.

In [26] Nešetřil and Ossona de Mendez described a linear-time algorithm that computes the partition $\{V_1, \dots, V_N\}$. The number of sets V_α where α is a subset of $[N]$ of size p is bounded by N^p. Then the number of graphs $G[V_\alpha]$ is bounded by N^p. Then the labeling J is constructed in linear-time since each labeling J_α is constructed in linear-time by Theorem 2.4.

We now examine the time taken to check whether G satisfies $\varphi(a_1, \dots, a_m)$. Each vertex x is in less than $(N-1)^{p-1}$ sets V_α. By comparing the sets that contain all the a_i's with the sets that contain a_1 we can determine in time $O(\log(n) \cdot |\bar{a}|)$ the sets V_α that contain $(a_1, \dots, a_m)$. For each V_α and each W_i we can determine in time $O(\log(n) \cdot |W_i|)$ the set $W_i \cap V_\alpha$. By Theorem 2.4 we can verify in each $G[V_\alpha]$ in time $O(\log(n) \cdot (|\bar{a}| + |\overline{W}| + 1))$ whether $G[V_\alpha]$ satisfies $\varphi(a_1, \dots, a_m, W_1 \cap V_\alpha, \dots, W_q \cap V_\alpha)$ since each $G[V_\alpha]$ has bounded tree-width. Therefore $\mathcal{B}$ checks the validity of $\varphi(a_1, \dots, a_m, W_1, \dots, W_q)$ in time $O(\log(n) \cdot (|\bar{a}| + |\overline{W}| + 1))$. $\square$

Proof of Theorem 5.1 (3). Let $\mathcal{C}$ be a locally cwd-decomposable class of graphs and let G in $\mathcal{C}$ be a graph with n vertices, represented by the structure $\langle V_G, (edg_{aG})_{a \in C_2}, (p_{aG})_{a \in C_1} \rangle$. Let $\varphi(\bar{x}, Y_1, \dots, Y_q)$ be a t-local formula around $\bar{x} = (x_1, \dots, x_m)$, $m \geq 1$. Then $G \models \varphi(\bar{a}, W_1, \dots, W_q)$ iff $G[N_G^t(\bar{a})] \models \varphi(\bar{a}, W_1 \cap N_G^t(\bar{a}), \dots, W_q \cap N_G^t(\bar{a}))$. Let Δ be a t-distance type with connected components $\Delta_1, \dots, \Delta_p$. By Lemma 3.5, $G \models \rho_{t,\Delta}(\bar{a}) \wedge \varphi(\bar{a}, W_1, \dots, W_q)$ iff $G \models \rho_{t,\Delta}(\bar{a}) \wedge F^{t,\Delta}(\varphi_{1,1}(\bar{a} \mid \Delta_1, W_1, \dots, W_q), \dots, \varphi_{p,j_p}(\bar{a} \mid \Delta_p, W_1, \dots, W_q))$.

We let $\mathcal{T}$ be an (r, ℓ, g)-cwd cover of G where $r = m(2t+1)$. We use this integer r to warranty that if $\Delta = \Delta(a_1, \dots, a_m)$ and $i_1, \dots, i_k$ in $[m]$ belong to a connected component of Δ then, $N_G^t(\{a_{i_1}, \dots, a_{i_k}\}) \subseteq U$ for some U in $\mathcal{T}$. This is so because $d_G(a_{i_1}, a_{i_{k'}}) \leq (m-1) \cdot (2t+1)$ for every $k' = 2, \dots, k$, hence, if $a \in N_G^t(\{a_{i_1}, \dots, a_{i_k}\})$ we have $d_G(a_{i_1}, a) \leq t + (m-1) \cdot (2t+1) \leq r$. Hence, $N_G^t(\{a_{i_1}, \dots, a_{i_k}\}) \subseteq N_G^r(a_{i_1}) \subseteq U$ for some U in $\mathcal{T}$. For each vertex x there exist less than ℓ many sets V in $\mathcal{T}$ such that $x \in V$. We assume that each set U in $\mathcal{T}$ has an index encoded as a bit string denoted by $\ulcorner U \urcorner$. There are at most $n \cdot \ell$ sets in $\mathcal{T}$. Hence $\ulcorner U \urcorner$ has length $O(\log(n))$.

For each set U in $\mathcal{T}$ we label each vertex in $G[U]$ with a label $K_U(x)$ of length $O(\log(n))$ in order to decide if $d_{G[U]}(x, y) \leq 2t+1$ or not by using $K_U(x)$ and

$K_U(y)$ [8] (Theorem 2.4). For each vertex x of G we let

$$K(x) = \Big(\ulcorner x \urcorner,\ \{\big(\ulcorner U \urcorner, K_U(x)\big) \mid N(x) \subseteq U\},\ \{\big(\ulcorner U \urcorner, K_U(x)\big) \mid N(x) \nsubseteq U\}\Big)$$

where $N(x) = N_G^{2t+1}(x)$. (We have $x \in N_G^t(x)$ for all $t \in \mathbb{N}$.) It is clear that $|K(x)| = O(\log(n))$.

By Theorem 2.4 for each formula $\varphi_{i,j}(\bar{x} \mid \Delta_i, Y_1, \ldots, Y_q)$ arising from Theorem 3.5 and each $U \in \mathcal{T}$ we can label each vertex $x \in U$ by some label $J^{\Delta}_{i,j,U}(x)$ of length $O(\log(n))$ so that we can decide if $\varphi_{i,j}(\bar{a} \mid \Delta_i, W_1, \ldots, W_q)$ holds in $G[U]$ by using $\big(J^{\Delta}_{i,j,U}(b)\big)_{b \in \bar{a}\ \mid \Delta_i}$ and $J^{\Delta}_{i,j,U}(W_1 \cap U), \ldots, J^{\Delta}_{i,j,U}(W_q \cap U)$. For each vertex x of G we let

$$J_\Delta(x) := \Big(\big(\ulcorner U \urcorner, J^{\Delta}_{1,1,U}(x), \ldots, J^{\Delta}_{1,j_1,U}(x), \ldots, J^{\Delta}_{p,1,U}(x), \ldots, J^{\Delta}_{p,j_p,U}(x)\big) \mid N_G^t(x) \subseteq U\Big).$$

It is clear that $|J_\Delta(x)| = O(\log(n))$ since each x is in less than ℓ many sets V in $\mathcal{T}$. There exist at most $k' = 2^{k(k-1)/2}$ t-distance type graphs; we enumerate them by $\Delta^1, \ldots, \Delta^{k'}$. For each vertex x of G we let $J(x) := (\ulcorner x \urcorner, K(x), J_{\Delta^1}(x), \ldots, J_{\Delta^{k'}}(x))$. It is clear that $J(x)$ is of length $O(\log(n))$.

By hypothesis the cover $\mathcal{T}$ is computed in time $f(n)$ for G in $\mathcal{C}$ with n vertices. By Theorem 2.4 the labelings K_U and $J^{\Delta}_{i,j,U}$ can be constructed in cubic-time. Therefore, the labeling J is constructed in time $O(f(n) + n^4)$ since there are less than $\ell \cdot n$ sets U in $\mathcal{T}$.

We now explain how to decide whether $G \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q)$ by using $J(a_1), \ldots, J(a_m)$ and $J(W_1), \ldots, J(W_q)$.

From the labels $K(x)$, we can determine the set $\{\ulcorner U \urcorner \mid U \in \mathcal{T},\ x \in U\}$, hence the family of sets $U \in \mathcal{T}$ such that $W \cap U \neq \emptyset,\ W \subseteq V_G$, where W is a set argument.

Since for each vertex x of G there exists a set U in $\mathcal{T}$ such that $N_G^r(x) \subseteq U$, for each pair of vertices (x, y) we have $d_G(x, y) \leq 2t + 1$ if and only if $d_{G[U]}(x, y) \leq 2t + 1$. Hence, by using the components $K(a_1), \ldots, K(a_m)$ from $J(a_1), \ldots, J(a_m)$ we can construct the t-distance type Δ of $(a_1, \ldots, a_m)$; let $\Delta_1, \ldots, \Delta_p$ be the connected components of Δ. From each $J(a_i)$ we can recover $J_\Delta(a_i)$. For each $\bar{a} \mid \Delta_i$ there exists at least one $U \in \mathcal{T}$ such that $N_G^t(\bar{a} \mid \Delta_i) \subseteq U$. We can determine these sets (there are less than ℓ of them) by using the labels in $J(b),\ b \in \bar{a} \mid \Delta_i$. We can now decide whether $G \models F^{t,\Delta}(\varphi_{1,1}(\bar{a} \mid \Delta_1, W_1 \cap U_1, \ldots, W_q \cap U_1), \ldots, \varphi_{p,j_p}(\bar{a} \mid \Delta_p, W_1 \cap U_p, \ldots, W_q \cap U_p))$ for some $U_1, \ldots, U_p$ determined from $J(a_1), \ldots, J(a_m)$. By using also $J(W_1), \ldots, J(W_q)$ we can determine the sets $W_i \cap U_j$ and this is sufficient by Theorem 3.5.

[8] For checking if $d_G(x, y) \leq 2t + 1$, an (r', ℓ', g')-cwd cover suffices, with $r' = 2t + 1$.

We now examine the time taken to check $\varphi(\bar{a}, \overline{W})$. For each couple (a_i, a_j) it takes time $O(\log(n))$ to check if $d(a_i, a_j) \leq 2t+1$. Since there are at most $|\bar{a}|^2$ couples, we construct the graph Δ in time $O(\log(n) \cdot |\bar{a}|^2)$. For each connected component $\bar{a} \mid \Delta$ we can determine the sets U that contain it in time $O(\log(n) \cdot |\bar{a}|)$ (less than ℓ such sets). By Theorem 2.4 we can check each $\varphi_{i,j}$ in time $O(\log(n) \cdot (|\bar{a}| + |\overline{W}| + 1))$. Therefore, $\mathcal{B}$ checks the validity of $\varphi(\bar{a}, \overline{W})$ in time $O(\log(n) \cdot (|\bar{a}|^2 + |\overline{W}| + 1))$. $\square$

Proof of Theorem 5.1 (4). Let $\mathcal{C}$ be a locally cwd-decomposable class of graphs and let G in $\mathcal{C}$ be a graph with n vertices, represented by the structure $\langle V_G, (edg_{aG})_{a \in C_2}, (p_{aG})_{a \in C_1} \rangle$. Let $\varphi(x_1, \ldots, x_m)$ be an FO formula without set arguments. By Theorem 3.4 the formula φ is equivalent to a Boolean combination $B(\varphi_1(\bar{x}), \ldots, \varphi_p(\bar{x}), \psi_1, \ldots, \psi_h)$ where φ_i is a t-local formula and ψ_i is a basic (t', s)-local sentence without set variables, for some t, t', s.

By Lemma 4.5 one can decide the validity of each sentence ψ_i. Let $b = (b_1, \ldots, b_h)$ where $b_i = 1$ if G satisfies ψ_i and 0 otherwise. For each $1 \leq i \leq p$ we construct a labeling J_i supporting query φ_i by Theorem 5.1 (3) (G belongs to a locally cwd-decomposable class and φ_i is a t-local formula around $\bar{x}$). For each vertex x of G we let $J(x) := (\ulcorner x \urcorner, J_1(x), \ldots, J_p(x), b)$. It is clear that $|J(x)| = O(\log(n))$ since $|J_i(x)| = O(\log(n))$. We now explain how to decide whether $G \models \varphi(a_1, \ldots, a_m)$ by using $J(a_1), \ldots, J(a_m)$.

From b we can recover the truth value of each sentence ψ_i. By using $J_i(\bar{a})$ we can check if $\varphi_i(\bar{a})$ holds. Then, we can check if $B(\varphi_1(\bar{x}), \ldots, \varphi_p(\bar{x}), \psi_1, \ldots, \psi_h)$ holds hence, if $\varphi(\bar{a})$ holds.

By Lemma 4.5 the validity of each sentence ψ_i is checked in time $O(n^4)$. And, by Theorem 5.1 (3), each labeling J_i is constructed in time $O(f(n) + n^4)$ where $f(n)$ is the time taken for constructing an (r, ℓ, g)-cwd cover. Hence, the labeling J can be constructed in time $O(f(n) + n^4)$. The time taken to check the validity of $\varphi(a_1, \ldots, a_m)$ is done in time $O(\log(n) \cdot |\bar{a}|^2)$ by Theorem 5.1 (3). $\square$

Before proving Theorem 5.1 (5) we introduce some definitions and facts. If $\mathcal{T}$ is an (r, ℓ, g)-cwd cover of a graph G, then $G(\mathcal{T})$ has maximum degree at most ℓ. Let m be a positive integer, a *proper distance-m coloring* of a graph H is a proper coloring of H^m (see Section 2 for the definition of H^m). Then, in a proper distance-m coloring, vertices at distance at most m have different colors. A graph G admits a proper $(d+1)$-coloring if d is its maximum degree. The graph $G(\mathcal{T})$ has maximum degree at most ℓ, hence, has a proper distance-m coloring with $\ell^{O(m)}$ colors since $G(\mathcal{T})^m$ has maximum degree at most $\ell \cdot (1 + (\ell - 1) + \cdots + (\ell - 1)^{m-1})$.

If $\mathcal{T}$ is cover of a graph G, for each positive integer t and each set U in $\mathcal{T}$ we let $K^t(U)$ be the set $\{x \in U \mid N^t_G(x) \subseteq U\}$. We call it the *t-kernel* of U.

Proof of Theorem 5.1 (5). Let $\mathcal{C}$ be a nicely locally cwd-decomposable class of graphs and let G in $\mathcal{C}$ be a graph with n vertices, represented by the structure $\langle\, V_G, (edg_{aG})_{a\in C_2}, (p_{aG})_{a\in C_1}\,\rangle$. We want a labeling for an FO query with set arguments. By Theorems 3.4 and 5.1 (3) it is sufficient to define a labeling for FO formulas $\varphi(Y_1,\ldots,Y_q)$ of the form:

$$\exists x_1 \cdots \exists x_m \left(\bigwedge_{1\le i<j\le m} d(x_i,x_j) > 2t \;\wedge\; \bigwedge_{1\le i\le m} \psi(x_i,Y_1,\ldots,Y_q) \right)$$

where $\psi(x,Y_1,\ldots,Y_q)$ is t-local around x. We show how to check such formulas by means of log-labelings.

We consider for purpose of clarity the particular case where $m=2$. Let $\mathcal{T}$ be a nice (r,ℓ,g)-cwd cover of G where $r=2t+1$, and let γ be a distance-2 coloring of $G(\mathcal{T})$, the intersection graph of $\mathcal{T}$ (vertices at distance 1 or 2 have different colors). For every two colors i and j we let $G_{i,j}$ be the graph induced by the union of the sets U in $\mathcal{T}$ that are colored by i or j (we may have $i=j$).

Claim 5.2 *$cwd(G_{i,j}) \le g(2)$.*

Proof of Claim 5.2. Let $\mathcal{T}^2 = \{U\cup U' \mid U,U' \in \mathcal{T},\ U\cap U' \neq \emptyset\}$. The graph $G_{i,j}$ is a disjoint union of sets in $\mathcal{T}\cup\mathcal{T}^2$. This union is disjoint because if $U\cup U'$ with $U\cap U'\neq\emptyset$ meets some $U''\in\mathcal{T}$ such that $U''\neq U$, $U''\neq U'$, then we have have $\gamma(U)=i$, $\gamma(U')=j\neq i$ and U'' meets U or U'. It can have neither color i nor color j because γ is a distance-2 coloring. Since $cwd(G[U\cup U']) \le g(2)$, we are done because the clique-width of a disjoint union of graphs $H_1,\ldots,H_s$ is $\max\{cwd(H_i) \mid i=1,\ldots,s\}$. $\square$

Claim 5.3 *Let $x\in K^{2t}(U)$ and $y \in K^{2t}(U')$ for some sets U and U' in $\mathcal{T}$. Then $d_G(x,y) > 2t$ iff $d_{G[U\cup U']}(x,y) > 2t$.*

Proof of Claim 5.3. The "only if direction" is clear since $d_G(x,y) \le d_{G[U\cup U']}(x,y)$.

For proving the converse, assume $d_G(x,y)\le 2t$; there exists a path of length at most $2t$ from x to y. This path is in $U\cup U'$ since $x\in K^{2t}(U)$ and $y\in K^{2t}(U')$. Hence it is also in $G[U\cup U']$, hence $d_{G[U\cup U']} \le 2t$. $\square$

Let us now give to each vertex x of G the smallest color i such that $x\in K^{2t}(U)$ and $\gamma(U)=i$. Hence each vertex has one and only one color. We express this by $p_i(x)$ where p_i is a new unary predicate. For each pair (i,j) (possibly $i=j$) we consider the formula $\psi_{i,j}$:

$$\exists x,y\Big(d(x,y)>2t \ \wedge\ \psi(x,Y_1,\ldots,Y_q)\ \wedge\psi(y,Y_1,\ldots,Y_q)\ \wedge\ p_i(x)\ \wedge\ p_j(y)\Big)$$

By Theorem 2.4 we can construct a *log*-labeling $J_{i,j}$ for the formula $\psi_{i,j}$ in the graph $G_{i,j}$. (We recall that vertex colors, i.e., additional unary relations, do not increase clique-width; the number of relations p_i does not depend on the graph G.) We compute the truth value $b_{i,j}$ of $\psi_{i,j}(\emptyset,\ldots,\emptyset)$ in $G_{i,j}$; we get a vector $\vec{b}$ of fixed length. We also label each vertex x by its color $\gamma(x)$. We concatenate that $\vec{b}$ and the $J_{i,j}(x)$ for $x\in V_{G_{i,j}}$, giving $J(x)$. The coloring γ uses $O(\ell^2)$ colors. Then, the number of graphs $G_{i,j}$ is bounded by $O(\ell^4)$. Therefore $|J(x)|=O(\log(n))$.

From $J(W_1),\ldots,J(W_q)$ we can determine those $G_{i,j}$ such that $V_{G_{i,j}}\cap(W_1\cup\cdots\cup W_q)\neq\emptyset$, and check if for one of them $G_{i,j}\ \models\ \psi_{i,j}(W_1,\ldots,W_q)$. If one is found we are done. Otherwise, we use the $b_{i,j}$'s to look for $G_{i,j}$ such that $G_{i,j}\ \models\ \psi_{i,j}(\emptyset,\ldots,\emptyset)$ and $(W_1\cup\cdots\cup W_q)\cap V_{G_{i,j}}=\emptyset$. This gives the correct results because of the following facts:

- If x,y satisfy the formula φ, then $x\in K^{2t}(U)$, $y\in K^{2t}(U')$ (possibly $U=U'$) and $d_G(x,y)>2t$ implies $d_{G_{i,j}}(x,y)>2t$, hence $G_{i,j}\ \models\ \psi_{i,j}(W_1,\ldots,W_q)$ where $i=\gamma(U)$ and $j=\gamma(U')$.
- If $G_{i,j}\ \models\ \psi_{i,j}(W_1,\ldots,W_q)$ then we get $G\ \models\ \varphi(W_1,\ldots,W_q)$ by similar argument (in particular $d_{G_{i,j}}(x,y)>2t$ implies $d_{G[U\cup U']}(x,y)>2t$ which implies that $d_G(x,y)>2t$ by Claim 5.3).

For $m=1$, the proof is similar by using a proper distance-1 coloring γ and the graphs $G_{i,i}$ instead of the graphs $G_{i,j}$.

For the case $m>2$, the proof is the same: one takes for γ a distance-m proper coloring of the intersection graph, one considers graphs $G_{i_1,\ldots,i_m}$ defined as (disjoint) unions of sets $U_1\cup\cdots\cup U_m$ for $U_1,\ldots,U_m$ in $\mathcal{T}$, of respective colors $i_1,\ldots,i_m$ and $cwd(G[U_1\cup\cdots\cup U_m])\leq g(m)$.

By hypothesis, the cover $\mathcal{T}$ is computed in time $f(n)$ for an n-vertex graph G in $\mathcal{C}$. In each graph $G_{i_1,\ldots,i_m}$ the labeling $J_{i_1,\ldots,i_m}$ is constructed in cubic-time by Theorem 2.4. The coloring γ uses $\ell^{O(m)}$ colors. Then, the number of graphs $G_{i_1,\ldots,i_m}$ is bounded by $\ell^{O(m^2)}$. Hence, the labeling J is computed in time $O(f(n)+n^3)$.

We now examine the time taken to check the validity of $\varphi(\overline{W})$. For each $G_{i_1,\ldots,i_m}$ and each W_i it takes time $O(\log(n)\cdot|W_i|)$ to determine $W_i\cap V_{G_{i_1,\ldots,i_m}}$. By Theorem 2.4 it takes time $O(\log(n)\cdot(|\overline{W}|+1))$ to check in $G_{i_1,\ldots,i_m}$ the validity of $\varphi(\overline{W})$. This terminates the proof of Theorem 5.1. $\Box$

Let us ask a very general question: what can be done with labels of size $O(\log(n))$? Here is a fact that limits the extension of these results.

Let $\varphi_0(x, y)$ be the t-local and bounded FO formula telling us whether two distinct vertices x and y are connected by a path of length 2:

$$x \neq y \wedge \exists z\,(z \neq x \wedge z \neq y \wedge edg(x, z)\ \wedge edg(z, y))$$

The adjacency query has a log-labeling scheme for graphs of bounded arboricity (Theorem 5.1 (1)).

Proposition 5.4 *Every labeling scheme supporting φ_0 on graphs with n vertices and of arboricity at most 2 requires labels of length at least $\sqrt{\frac{n}{2}} - 1$ for some graphs.*

Proof. With every simple, loop-free and undirected graph G we associate the graph $\widetilde{G}$ obtained by inserting a vertex z_{xy} on each edge xy.

$$V_{\widetilde{G}} = V_G \cup \{z_{x,y} \mid x, y \in V_G \text{ and } xy \in E_G\},$$
$$E_{\widetilde{G}} = \{x z_{x,y} \mid xy \in E_G\}.$$

The following properties hold.

(1) $V_G \subseteq V_{\widetilde{G}}$ and $|V_{\widetilde{G}}| = |V_G| + |E_G|$.
(2) For all $x, y \in V_G$, $xy \in E_G$ if and only if $\widetilde{G} \models \varphi_0(x, y)$.
(3) $\widetilde{G}$ has arboricity at most 2.

The first two points are clear. For the third one we orient each edge e of G and we get a directed graph, that we denote by $\vec{G}$. We let:

$$F_1 = \{x z_{x,y} \mid (x, y) \in E_{\vec{G}}\},$$
$$F_2 = \{z_{x,y} y \mid (x, y) \in E_{\vec{G}}\}.$$

Neither F_1 nor F_2 has a cycle in $\widetilde{G}$. Then $\widetilde{G}$ has arboricity at most 2 since (F_1, F_2) is a bipartition of $E_{\widetilde{G}}$.

By using a simple counting argument, one can show that every labeling scheme supporting adjacency in simple and undirected graphs with n vertices requires some labels of size at least $\frac{1}{n}\log_2\left(2^{\binom{n}{2}}\right) = (n-1)/2$ bits. Hence, adjacency requires labels of size $\lfloor n/2 \rfloor$ in all graphs. Using (2) above, we conclude that any labeling scheme for φ_0 on the graph family $\mathcal{F}_n = \{\widetilde{G} \mid G \text{ has } n \text{ vertices}\}$ requires labels of size at least $\left\lfloor \frac{n}{2} \right\rfloor$. Let $\widetilde{G}$ be in $\mathcal{F}_n$ and let $\tilde{n} = |V_{\widetilde{G}}|$. Using (1) we have $\tilde{n} = n + |E_G| \leq \frac{n(n+1)}{2}$, i.e., $n \geq \sqrt{2\tilde{n}} - 1$. Hence, any labeling scheme for φ_0 on $\mathcal{F}_n$ requires for some graphs with $\tilde{n}$ vertices labels of size at least $\left\lfloor \frac{\sqrt{2\tilde{n}}-1}{2} \right\rfloor > \sqrt{\frac{\tilde{n}}{2}} - 1$. $\square$

6 Extension to Counting Queries

We now consider an extension to counting queries.

Definition 6.1 (Counting Query) *Let $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be an FO or MSO formula and let G be a (colored) graph. For $W_1, \ldots, W_q \subseteq V_G$ we let:*

$$\#_G\varphi(W_1, \ldots, W_q) := \left| \left\{ (a_1, \ldots, a_m) \in V_G^m \mid G \models \ \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q) \right\} \right|.$$

The counting query of φ *consists in determining $\#_G\varphi(W_1, \ldots, W_q)$ for given $(W_1, \ldots, W_q)$. If $s \geq 2$ the* counting query of φ modulo s *consists in determining $\#_G\varphi(W_1, \ldots, W_q)$ modulo s for given $(W_1, \ldots, W_q)$.*

The following theorem is an easy extension of Theorem 2.4.

Theorem 6.2 *Let k be a positive integer and, let $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be an MSO formula over colored graphs (binary relational structures) and $s \geq 2$. There exists a* $\log^2$*-labeling scheme (resp. a* $\log$*-labeling scheme) $(\mathcal{A}, \mathcal{B})$ on the class of graphs of clique-width at most k for the counting query of φ (resp. the counting query of φ modulo s). Moreover, if the input graph has n vertices then, algorithm $\mathcal{A}$ constructs the labels in time $O(n^3)$ or in $O(n \cdot \log(n))$ if the clique-width expression is given; algorithm $\mathcal{B}$ computes $\#_G\varphi(W_1, \ldots, W_q)$ in time $O(\log^2(n) \cdot (|\overline{W}| + 1))$ (resp. $O(\log(n) \cdot (|\overline{W}| + 1))$).*

We will prove a similar theorem for nicely locally cwd-decomposable classes of graphs and FO formulas.

Theorem 6.3 (Second Main Theorem) *Let $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be an FO formula and let $s \geq 2$. There exists a* $\log^2$*-labeling scheme (resp. a* $\log$*-labeling scheme) $(\mathcal{A}, \mathcal{B})$ for the counting query of φ (resp. the counting query of φ modulo s) on nicely locally cwd-decomposable classes. Moreover, if the input graph has n vertices then, algorithm $\mathcal{A}$ constructs the labels in time $O(f(n) + n^3)$ where $f(n)$ is the time taken to construct a nice cwd-cover; algorithm $\mathcal{B}$ computes $\#_G\varphi(W_1, \ldots, W_q)$ in time $O(\log^2(n) \cdot (|\overline{W}| + 1))$ (resp. $O(\log(n) \cdot (|\overline{W}| + 1))$).*

We will first prove Theorem 6.3 for particular t-local formulas on locally cwd-decomposable classes.

Definition 6.4 (t-Connected Formulas) *A formula $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ is t-*connected *if for all G, $a_1, \ldots, a_m \in V_G$ and $W_1, \ldots, W_q \subseteq V_G$,*

$$G \models \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q) \quad \textit{iff} \quad \begin{cases} \bigwedge_{1 \leq i < j \leq m} d(a_i, a_j) \leq t \ \textit{and} \\ G[N] \models \varphi(a_1, \ldots, a_m, W_1 \cap N, \ldots, W_q \cap N) \end{cases}$$

where $N = N_G^t(\{a_1, \ldots, a_m\})$.

Remark 6.5 *Let $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be a t-connected formula. Then for all $W \supseteq N_G^t(a_1, \ldots, a_m)$:*

$$G \models \ \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q) \quad \textit{iff} \quad G[W] \models \ \varphi(a_1, \ldots, a_m, W_1 \cap W, \ldots, W_q \cap W)$$

and, since $N_G^t(\{a_1, \ldots, a_m\}) \subseteq N_G^{2t}(a_1)$, we have

$$G \models \ \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q) \quad \textit{iff} \quad G[N_G^{2t}(a_1)] \models \ \varphi(a_1, \ldots, a_m, W_1, \ldots, W_q).$$

Lemma 6.6 *Let $\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q)$ be a t-connected formula and let $s \geq 2$. Then, there exists a $\log^2$-labeling scheme (resp. a $\log$-labeling scheme) $(\mathcal{A}, \mathcal{B})$ for the counting query of φ (resp. the counting query of φ modulo s) on locally cwd-decomposable classes of graphs. Moreover, if the input graph has n vertices then, algorithm $\mathcal{A}$ constructs the labels in time $O(f(n) + n^3)$ where $f(n)$ is the time taken to construct a cwd-cover; algorithm $\mathcal{B}$ computes $\#_G\varphi(W_1, \ldots, W_q)$ in time $O(\log^2(n) \cdot (|\overline{W}| + 1))$ (resp. $O(\log(n) \cdot (|\overline{W}| + 1))$).*

Proof. Let $\mathcal{C}$ be a locally cwd-decomposable class of graphs and let $\mathcal{T}$ be a $(2t, \ell, g)$-cwd cover of an n-vertex graph G from $\mathcal{C}$. Let H be the intersection graph of $\mathcal{T}$ (Definition 4.2) and let γ be a proper coloring of H with colors in $[\ell + 1]$.

Claim 6.7 *Let $x \in K_G^{2t}(U)$ and $y \in U'$ with $\gamma(U) = \gamma(U')$, $U \neq U'$. Then $d_G(x, y) > 2t$.*

Proof of Claim 6.7. If this is not the case, then $y \in U$ and x_U and $x_{U'}$ are adjacent in H. This is impossible since they have the same color. □

We color each vertex x of G by i, the smallest $\gamma(U)$ such that $x \in K_G^{2t}(U)$. We represent this by the validity of $p_i(x)$, as in the proof of Theorem 5.1 (5). For each $i \in [\ell + 1]$ we let φ_i be the formula:

$$\varphi(x_1, \ldots, x_m, Y_1, \ldots, Y_q) \ \wedge \ p_i(x_1).$$

Then the following is clear.

Claim 6.8 $\#_G\varphi(Y_1,\ldots,Y_q) = \sum_i \#_G\varphi_i(Y_1,\ldots,Y_q)$.

We now show that the counting query of φ admits a $\log^2$-labeling scheme on G. We let $V_i = \bigcup_{\gamma(U)=i} \{U \mid U \in \mathcal{T}\}$.

Claim 6.9 $cwd(G[V_i]) \leq g(1)$.

Proof of Claim 6.9. V_i is a disjoint union of sets U from $\mathcal{T}$. From Definition 4.3 each graph $G[U]$ has clique-width at most $g(1)$. Therefore $cwd(G[V_i]) \leq g(1)$. $\square$

Claim 6.10 $\#_G\varphi_i(Y_1,\ldots,Y_q) = \#_{G[V_i]}\varphi_i(Y_1,\ldots,Y_q)$.

Proof of Claim 6.10. If $\varphi(a_1,\ldots,a_m,W_1,\ldots,W_q)$ holds and $p_i(a_1)$ holds then, $a_1 \in K_G^{2t}(U)$ for some U such that $\gamma(U) = i$. Hence $a_2,\ldots,a_m \in N_G^{2t}(a_1)$ and $G[N_G^{2t}(a_1)] \models \varphi_i(a_1,\ldots,a_m,W_1,\ldots,W_q)$, hence $G[V_i] \models \varphi_i(a_1,\ldots,a_m, W_1,\ldots,W_q)$.

If $G[V_i] \models \varphi_i(a_1,\ldots,a_m,W_1,\ldots,W_q)$, then $p_i(a_1)$ holds and $d_{G[V_i]}(a_l,a_s) \leq t$ for all $l,s \in [m]$. But $d_G(a_l,a_s) = d_{G[V_i]}(a_l,a_s) = d_{G[U]}(a_l,a_s)$ where $a_1 \in U$ and $\gamma(U) = i$. And since $N_G^t(\{a_1,\ldots,a_m\}) \subseteq V_i$ we have $G \models \varphi_i(a_1,\ldots,a_m, W_1,\ldots,W_q)$. $\square$

By Theorem 6.2 and Claims 6.9 and 6.10 there exists a $\log^2$-labeling J_i for the counting query of each φ_i. For each $x \in V_G$ we let $J(x) = (J_1(x),\ldots,J_{\ell+1}(x))$. Hence J is a $\log^2$-labeling for the counting query of φ by Claim 6.8. By Theorem 6.2 labels of size $O(\log(n))$ are sufficient for the counting query of each φ_i modulo s.

By Theorem 6.2 each labeling J_i is constructed in cubic-time. Therefore, the labeling J is constructed in time $O(f(n)+n^3)$ where $f(n)$ is the time taken for constructing the $(2t,\ell,g)$-cwd cover $\mathcal{T}$ of G. By Claim 6.8 and Theorem 6.2 $\mathcal{B}$ computes $\#_G\varphi(W_1,\ldots,W_q)$ in time $O(\log^2(n)\cdot(|\overline{W}|+1))$ (resp. $O(\log(n)\cdot (|\overline{W}|+1))$). $\square$

We now prove Theorem 6.3.

Proof of Theorem 6.3. Let $\varphi(\bar{x},\overline{Y})$ be an FO formula with free variables in $\bar{x} = (x_1,\ldots,x_m)$ and in $\overline{Y} = (Y_1,\ldots,Y_q)$. By Theorem 3.4 φ is logically equivalent to a Boolean combination of t-local formulas around $\bar{x}$ and of basic (t', s)-local formulas. We have proved that each basic (t', s)-local formula admits a log-labeling scheme on each nicely locally cwd-decomposable class of graphs (Theorem 5.1 (5)). It remains to prove that the counting query of a t-local formula admits a $\log^2$-labeling scheme on each nicely locally cwd-decomposable class of graphs $\mathcal{C}$. Let G, a graph with n vertices, be in $\mathcal{C}$

Let $\psi(\bar{x}, Y_1, \ldots, Y_q)$ be a t-local formula around $\bar{x} = (x_1, \ldots, x_m)$. By Theorem 3.5 we can reduce the counting query of ψ to the counting query of finitely many formulas of the form $\rho_{t,\Delta}(\bar{x}) \ \wedge \ \varphi'(\bar{x}, Y_1, \ldots, Y_q)$ that can be expressed as

$$\varphi'(\bar{x}, Y_1, \ldots, Y_q) := \bigwedge_{1 \le i < j \le p} d(\bar{x} \mid \Delta_i, \bar{x} \mid \Delta_j) > 2t + 1 \ \wedge \ \bigwedge_{1 \le i \le p} \varphi_i(\bar{x} \mid \Delta_i, Y_1, \ldots, Y_q)$$

where each φ_i is t-local and $(m \cdot (2t + 1))$-connected. We can assume that ψ is of the form $\varphi'(\bar{x}, Y_1, \ldots, Y_q)$.

Let $\mathcal{T}$ be a nice (r, ℓ, g)-cwd cover where $r = m \cdot (2t+1)$ and let γ be a proper distance-m coloring of $G(\mathcal{T})$, the intersection graph of $\mathcal{T}$. For every m-tuple of colors $(i_1, \ldots, i_m)$ we let $G_{i_1,\ldots,i_m}$ be the graph $G[V]$ where V is the union of all sets $U \in \mathcal{T}$ such that $\gamma(U) \in \{i_1, \ldots, i_m\}$. We have then $cwd(G[V]) \le g(m)$ (same arguments as in Claim 5.2). We color each vertex with the smallest color i such that $x \in K_G^r(U)$ and $\gamma(U) = i$ and we express this by the validity of $p_i(x)$. We let $\varphi'_{i_1,\ldots,i_m}$ be

$$\bigwedge_{1 \le i < j \le p} d(\bar{x} \mid \Delta_i, \bar{x} \mid \Delta_j) > 2t + 1 \ \wedge \ \bigwedge_{1 \le \ell \le p} (\varphi_\ell(\bar{x} \mid \Delta_\ell, Y_1, \ldots, Y_q) \ \wedge \ p_{i_\ell}(z_\ell))$$

where z_ℓ is the first variable of each tuple $\bar{x} \mid \Delta_\ell$. We have:

Claim 6.11 $\#_G\psi(Y_1, \ldots, Y_q) = \sum_{(i_1,\ldots,i_m)} \#_G\varphi'_{i_1,\ldots,i_m}(Y_1, \ldots, Y_m)$.

We let $H = G_{i_1,\ldots,i_m}$. By the same arguments as in the proof of Claim 5.3 we have:

Claim 6.12 $d_G(\bar{x} \mid \Delta_i, \bar{x} \mid \Delta_j) > 2t + 1$ *if and only if* $d_H(\bar{x} \mid \Delta_i, \bar{x} \mid \Delta_j) > 2t + 1$.

It follows that:

Claim 6.13 $\#_G\varphi'_{i_1,\ldots,i_m}(Y_1, \ldots, Y_q) = \#_H\varphi'_{i_1,\ldots,i_m}(Y_1, \ldots, Y_q)$.

By Theorem 6.2 and Claims 6.11, 6.12 and 6.13 there exists a $\log^2$-labeling scheme for the counting query of each t-local formula, and a log-labeling scheme is enough for modulo counting.

By hypothesis, a nice (r, ℓ, g)-cwd cover $\mathcal{T}$ of G can be constructed in time $f(n)$. For each formula $\varphi_{i_1,\dots,i_m}$ the associated labeling $J_{i_1,\dots,i_m}$ is constructed in time $O(n^3)$ by Theorem 6.2. The coloring γ uses $\ell^{O(m)}$ colors. The number of graphs $G_{i_1,\dots,i_m}$ is bounded by $\ell^{O(m^2)}$. Hence, the labeling J is computed in time $O(f(n) + n^3)$. By Claim 6.11 and Theorem 6.2 algorithm $\mathcal{B}$ computes $\#_G\varphi(W_1, \dots, W_q)$ in time $O(\log^2(n) \cdot (|\overline{W}| + 1))$ (resp. $O(\log(n) \cdot (|\overline{W}| + 1))$). This finishes the proof. $\square$

7 Conclusion

We conjecture that the results of Theorem 5.1 (3-5) extend to classes of graphs that exclude, or locally exclude a minor (definitions are from [11,17]).

Question 1 *Does there exist a* log*-labeling scheme for FO formulas with set arguments on locally cwd-decomposable classes?*

References

[1] S. Arnborg, J. Lagergren and D. Seese. Easy Problems for Tree-Decomposable Graphs. *Journal of Algorithms* 12(2):308-340, 1991.

[2] G. Bagan. PhD Thesis, Université de Caen. March, 2009.

[3] A. Blumensath and B. Courcelle. Recognizability, Hypergraph Operations and Logical Types. *Information and Computation* 204(6):853-919, 2006.

[4] H.L. Bodlaender. A Linear-Time Algorithm for Finding Tree-Decompositions of Small Tree-width. *SIAM Journal on Computing* 25(6):1305-1317, 1996.

[5] H.L. Bodlaender. Tree-Width: Structure and Algorithms. In G. Principe and S. Zaks eds., *Structural Information and Communication Complexity (SIROCCO)*, volume 4474 of *LNCS*, pages 11-25. Springer, 2007.

[6] B. Courcelle, C. Gavoille, M.M. Kanté and A. Twigg. Optimal Labeling for Connectivity Checking in Planar Networks with Obstacles. Manuscript, 2008. An extended abstract will appear in *Electronic Notes in Discrete Mathematics*, proceedings of the first Conference *Topological and Geometric Graph Theory (TGGT)*, Paris, 2008.

[7] B. Courcelle, J.A. Makowsky and U. Rotics. Linear-Time Solvable Optimization Problems on Graphs of Bounded Clique-Width. *Theory of Computing Systems* 33(2):125-150, 2000.

[8] B. Courcelle and S. Olariu. Upper Bounds to the Clique-Width of Graphs. *Discrete Applied Mathematics* 101(1-3):77-114, 2000.

[9] B. Courcelle and A. Twigg. Compact Forbidden-Set Routing. In W. Thomas and P. Weil eds., *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 4393 of *LNCS*, pages 37-48. Springer, 2007.

[10] B. Courcelle and R. Vanicat. Query Efficient Implementation of Graphs of Bounded Clique-Width. *Discrete Applied Mathematics* 131(1):129-150, 2003.

[11] A. Dawar, M. Grohe and S. Kreutzer. Locally Excluding a Minor. In 22^{nd} *IEEE Symposium on Logic in Computer Science (LICS)*, pages 270-279. IEEE Computer Society, 2007.

[12] Arnaud Durand and Etienne Grandjean. First-Order Queries on Structures of Bounded Degree are Computable with Constant Delay. *ACM Transactions on Computational Logic* 8(4), 2007.

[13] E. Fisher, J.A. Makowsky and E.V. Ravve. Counting Truth Assignments of Formulas of Bounded Tree-Width or Clique-Width. *Discrete Applied Mathematics* 156(4):511-529, 2008.

[14] M. Frick. Generalized Model-Checking over Locally Tree-Decomposable Classes. *Theory of Computing Systems* 37(1):157-191, 2004.

[15] M. Frick and M. Grohe. Deciding First-Order Properties of Locally Tree-Decomposable Structures. *Journal of the ACM* 48(1):1184-1206, 2001.

[16] H. Gaifman. On Local and Non-Local Properties. In *Proceedings of the Herbrand Symposium Logic Colloquium'81* pages 105-135, 1982.

[17] M. Grohe. Logic, Graphs and Algorithms. In Flum, Grädel, Wilke eds., *Logic, Automata, History and Perspectives*, pages 357-422. Amsterdam University Press, 2007.

[18] C. Gavoille and D. Peleg. Compact and Localized Distributed Data Structures. *Distributed Computing* 16(2-3):111-120, 2003.

[19] F. Gurski and E. Wanke. Line Graphs of Bounded Clique-Width. *Discrete Mathematics* 307(22):2734-2754, 2007.

[20] P. Hliněný and S. Oum. Finding Branch-Decompositions and Rank-Decompositions. In L. Arge, M. Hoffmann and E. Welzl eds., *Annual European Symposium*, volume 4698 of *LNCS*, pages 163-174. Springer, 2007.

[21] M. Kamiński, V. Lozin and M. Milanič. Recent Developments on Graphs of Bounded Clique-Width. *Discrete Applied Mathematics*, in press, 2008.

[22] M. M. Kanté. *Graph Structurings: Some Algorithmic Applications*. PhD thesis, Université Bordeaux 1, Bordeaux, 2008.

[23] S. Kreutzer. Algorithmic Meta-Theorems. Manuscript, 2008.

[24] L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.

[25] V. Lozin. Clique-Width of Unit Interval Graphs. Manuscript, arXiv:0709.1935, 2007.

[26] J. Nešetřil and P. Ossona de Mendez. Linear Time Low Tree-Width Partitions and Algorithmic Consequences. In J.M. Kleinberg ed., 38^th *Annual ACM Symposium on Theory of Computing (STOC)*, pages 391-400. ACM, 2006.

[27] S. Oum and P. Seymour. Approximating Clique-Width and Branch-Width, *Journal of Combinatorial Theory, Series B*, 96(4):514-528, 2006.

[28] D. Seese. Linear Time Computable Problems and First-Order Descriptions. *Mathematical Structures in Computer Science* 6(6):505-526, 1996.